“Number” Class

Goal: This project is to implement a “Number” class and then create two kinds of Numbers: “Integer” and “Fraction” class.

Description: The Number class supports three operations: display, “==”, and “+”. The detailed description of these three operations is shown as follows.

1. **display**: This operation displays the Number itself in its original form.
2. **==:** This operation returns true if two Numbers are numerically the same.
3. **+:** This operation adds the Number itself with another Number and returns a third Number whose numeric value is equal to the sum of the numeric values of the former two.

The Integer class is represented by its integer value. The Fraction class is represented by its numerator and denominator (both are integers), and it needs to be displayed in its original form. That is, "2/4" has to be displayed as such, not "1/2" or "0.5".

Integer and Fraction are the only two kinds of Numbers. Naturally, they support all three operations of Number. To be more precise, you can add two Integers, or two Fractions, or one Integer and one Fraction.

Solution: Briefly, I created a pure abstract base class "Number" and then built the two kinds of Numbers: "Integer" and "Fraction", derived from the "Number" class. All the classes were implemented in a form of library. I used the factory pattern to create the objects for these two subclass so that the user of this library wouldn’t know specific “Numbers” except at the moment of construction of these Numbers. I also wrote a test file to test my solutions.

I created two solutions using C++14. Solution1 utilized raw pointers and supported cascading “+” operations. Solution2 applied smarter pointers but couldn’t support cascading “+” operations. For each solution, I provided two versions. One is for Linux system and the other is for Windows. I used Visual Studio 2017 to create the library and a test project for Windows version. For Linux version, I used gnu++11 to compile my solution and I implemented 'make_unique' in the source code due to lack of this function in gnu++11.

For more details, please refer to the readme.txt file under the corresponding directory.